NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

Survey of k-Anonymity

by

Ankit Saroha

A thesis submitted in partial fulfillment for the
degree of Bachelor of Technology

under the guidance of
Dr. K. S. Babu
Department of Computer Science & Engineering

March 2014

Certificate

This is to certify that the work in the thesis entitled **"Survey of k-Anonymity"** submitted by Ankit Saroha of department of computer science & engineering at National Institute of Technology Rourkela is a record of authentic work performed by him under my supervision and guidance in partial fulfillment for the degree of Bachelor of Technology in Computer Science & Engineering at National Institute of Technology, Rourkela. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree or diploma.

Signed:

(Dr. Korra Sathya Babu
Department of Computer Science & Engineering
National Institute of Technology Rourkela)

Date:

Abstract

Many organisations are releasing microdata everyday for business and research purposes. This data does not include explicit identifiers of an individual like name or address but it does contain data like date of birth, pin code, sex, marital-status etc which when combined with other publicly released data like voter registration data can identify an individual. This joining attack can also be used to obtain sensitive information about an individual, thus, putting the privacy of an individual in grave danger.

K-anonymization is a technique that prevents the above mentioned attacks by modifying the microdata which is released for business or research purposes. This is done by applying generalization and suppression techniques to the microdata. In this paper, k-anonymity is introduced and also some of the algorithms are studied which help in achieving k-anonymity.

Acknowledgements

I wish to express my sense of gratitude towards Dr. Korra Sathya Babu, Department of Computer Science & Engineering, National Institute of Technology, Rourkela, my guide, for giving me this wonderful opportunity to work with him and for his motivation, consistent encouragement, support and cooperation to carry out this project. This would not have been possible without his expert guidance. I am also grateful to the people whose works I referred to whose details are mentioned in the bibliography.

At last, I thank all my friends for their support who extended all sorts of help, when needed, in completing this project.

Contents

Chapter 1

Introduction

Microdata is being published by many organizations for many different purposes such as business, demographic research, public health research etc. This published data can put the privacy of an individual at risk. To protect the anonymity of the entities, the data holders encrypt or remove the explicit identifiers such as name, phone numbers, social security number and addresses.

However, other attributes like sex, date of birth, zip code, race etc when combined together with publicly released information, can be used to identify the anonymous individuals. The large amount of information that is easily accessible today, when combined with the increased computational power available to the attackers, make such attacks a serious problem.

Information about us is collected on a day to day basis, as we join companies or groups, shop for groceries, or execute our common daily activities the amount of privately owned records describing each citizen's finances, interests, and demographics is increasing every day. Information bureaus such as TRW, Equifax, and Trans Union hold the largest and most detailed databases on American consumers. Many municipalities sell population registers that include the identities of individuals along with basic demographics; examples include voter lists, city directories, local census data, tax assessors, information from motor vehicle agencies, and real estate agencies. Typical data contained in these databases may include names, social security numbers, race, date of birth, addresses, telephone numbers, marital status, and employment/salary histories. These data, which are often publicly distributed or sold, can be used for linking identities with de-identified information, thus allowing re-identification of respondents. This type of situation has raised particular concerns in the medical and financial fields, where microdata, which are increasingly released for circulation or research, can be or have been subject to abuses, threatening the privacy of individuals.

1.1 Objective

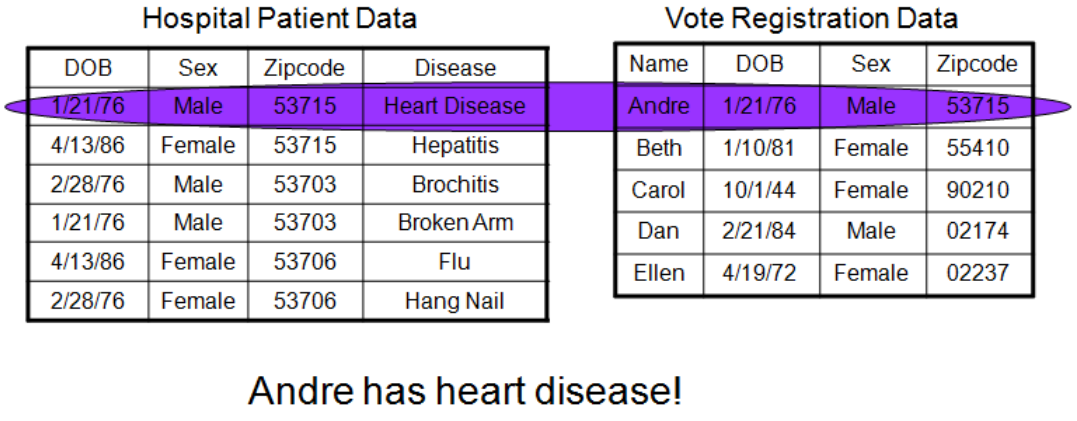
The following are the main objectives which need to be achieved:-

- To release maximum amount of data so that it can be used for business or research related work by various organisations

- To ensure that privacy of no individual is being put in danger due to the released data by protecting released information against inference and linking attacks

1.2 Wrong Approach

Removing the unique identifiers such as **Name, Employee_Id** from a table cannot guarantee privacy. Other attributes like **Date_of_Birth, Sex, PIN_Code** when combined together can also reveal the identity of an individual.Re-identification is possible by using a set of attributes and another database containing the same set of attributes. Sometimes this approach can also leak sensitive information about an individual. An example depicting the attack is shown below:



An attacker can simply combine the information obtained from hospital patient data and vote registration data. By matching the attributes like DOB, Sex and Zipcode the attacker can easily infer that Andre is suffering from a heart disease which is a very sensitive information related to an individual. As it is pretty evident that hiding the name, phone number or other explicit identifiers does not guarantee the security of sensitive

information of an individual, therefore, we need more effective techniques to achieve our objective.

1.3 Related Work

A few approaches that can solve our problem and help us in our objective are the following:-

- k-anonymity
- l-diversity
- t-closeness

Chapter 2

Literature Survey of k-Anonymity

k-anonymity is one of the techniques which help us in releasing a huge amount of data so that it can be used for business or research related work by various organisations by ensuring that privacy of no individual is being put in danger due to the released data by protecting released information against inference and linking attacks.

2.1 Basic Definitions

- **Key Attribute** - The attribute that can identify an individual directly is known as the *key attribute*. It is always removed during the release of data. e.g. Name, Mobile_No.
- **Quasi-Identifier** - The set of attributes that can be used to identify an individual by using any means is called *quasi-identifier*. e.g Date_of_Birth, PIN_Code.
- **Sensitive Attribute** - The attribute containing the sensitive information about an individual is the *sensitive attribute*. e.g. Salary, Health_Problem

2.2 k-Anonymity

k-anonymity states that there should be at least k tuples having the same quasi-identifier values to guarantee an individual's privacy. Every tuple in a table should be similar to at least (k-1) tuples then only the table will achieve k-anonymity.

K-anonymity is achieved by using generalization and suppression.Following is an example of a table satisfying 2-anonymity with respect to each attribute:-

Sex	PIN_Code	Salary
Male	110010	>50k
Female	110011	>100k
Male	110011	>100k
Female	110010	>50K
Male	110012	>150k
Female	110012	>150k

2-anonymous Table

2.3 Generalization

Conversion of any value to a more general form is the process of **generalization**. E.g. “Male” and “Female” can be generalized to “Person”. Generalization can be applied at the following levels:

- **Attribute (AG)**: Generalization is performed at the level of column; a generalization step generalizes all the values in the column.
- **Cell (CG)**: Generalization is performed on single cells; as a result a generalized table may contain, for a specific column, values at different generalization levels. E.g. DOB, where generalizing date, month and year form different levels of generalization.

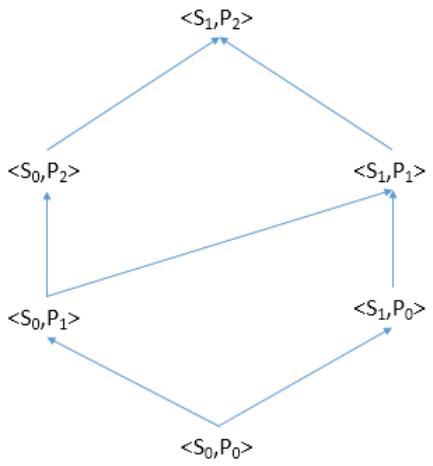
2.3.1 Domain Generalization Hierarchy

Domain Generalization Hierarchy can be defined as a graph or a lattice which acts as the solution space for our k-anonymity problem. The nodes of this lattice are achieved by generalizing different combination of attributes together at various levels.

Example

Consider two attributes ”Sex” and ”PIN_Code” of a relation T. Value of attribute Sex at level 0 of generalization can be ”Male” and ”Female”. To achieve level 1 of generalization with respect to attribute Sex we must generalize the values ”Male” and ”Female”. We can generalize these two values to another value, say, ”Person”. By generalizing the

values of attribute Sex to "Person" we achieve level 2 generalization with respect to Sex. Lets take another attribute PIN_Code from relation T. Let us assume that PIN_Code can have values "110010", "110011" and "110012" at level 0 generalization. We can generalize these values to "11000x" and "11001x" to achieve level 1 generalization with respect to attribute PIN_Code. Further, we can generalize the values to "1100xx" in order to achieve level 2 generalization with respect to attribute PIN_Code. By combining different levels of generalization of different attributes we can form the Domain Generalization Hierarchy in the following manner:-



Domain Generalization Hierarchy(DGH) with respect to attributes Sex and PIN_Code

2.4 Suppression

Removing any value completely from a data table is the process of suppression. Suppression can be applied at the following levels:

- **Tuple (TS)**: Suppression is performed at the level of row; a suppression operation removes a whole tuple
- **Attribute (AS)**: Suppression is performed at the level of column, a suppression operation obscures all the values of a column.
- **Cell (CS)**: Suppression is performed at the level of single cells; as a result a k-anonymized table may wipe out only certain cells of a given tuple/attribute.

2.5 Models of k-anonymity

he possible combinations of different types of generalizations and suppressions result in different models of k-anonymity. The following are the different models:-

- **AG_TS**: Generalization is applied at the level of attribute (column) and suppression at the level of tuple (row).
- **AG_AS**: Both generalization and suppression are applied at the level of column. No specific approach has investigated this model. It must also be noted that if attribute generalization is applied, attribute suppression is not needed. It becomes equivalent to AG_(attribute generalization and no suppression).
- **AG_CS**: Generalization is applied at the level of column, while suppression at the level of cell. It allows to reduce the effect of suppression, at the price however of a higher complexity of the problem.
- **AG_**: Generalization is applied at the level of column, suppression is not considered.
- **CG_CS**: Both generalization and suppression are applied at the cell level. Then, for a given attribute we can have values at different levels of generalization. By observations, this model is equivalent to CG_ (cell generalization, no suppression).
- **CG_**: Generalization is applied at the level of cell, suppression is not considered.
- **_TS**: Suppression is applied at the tuple level, generalization is not allowed.
- **_AS**: Suppression is applied at the attribute level, generalization is not allowed. No explicit approach has investigated this model.
- **_CS**: Suppression is applied at the cell level, generalization is not allowed. Again, it can be modeled as a reduction of AG_.

Chapter 3

k-Anonymity Algorithms and Their Comparison With Various Parameters

3.1 k-Anonymity Algorithms

Based on different models of k-anonymity there are various algorithms to achieve k-anonymity. A few algorithms for AG_TS model are the following:-

- **Incognito Algorithm**
- **Samarati's Algorithm**
- **Sweeney's Algorithm**

3.2 Incognito Algorithm

3.2.1 Basic Incognito

This algorithm produces all the possible k-anonymous full-domain generalizations of a relation(say T), with an optional tuple suppression threshold. It begins by checking single-attribute subsets of the quasi-identifier, and then iterates, checking k-anonymity with respect to larger subsets of quasi-identifiers.

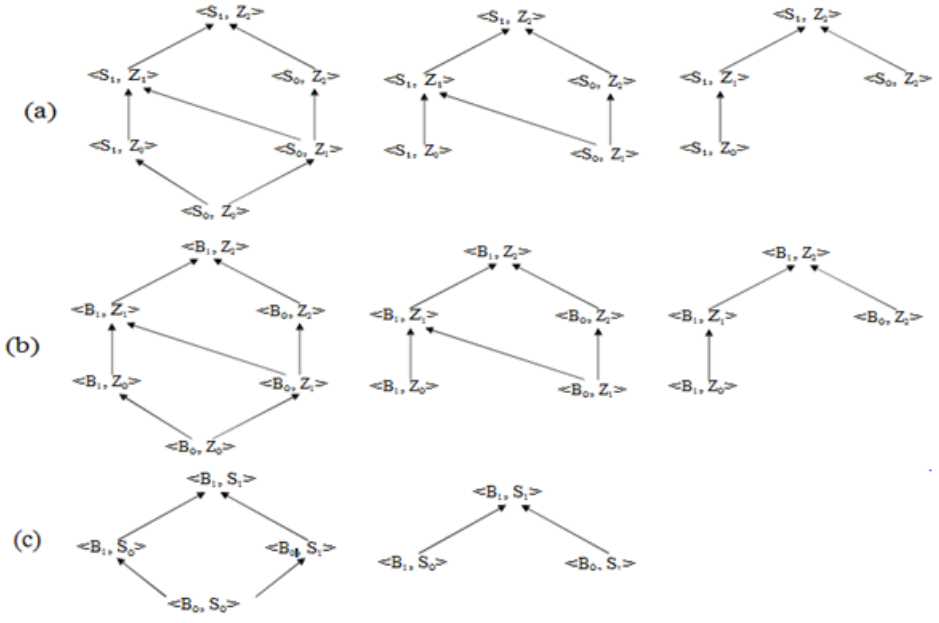
3.2.2 Algorithm

Input: Private table PT; quasi-identifier QI=(A_1,....,A_n), k constraint; hierarchies DGH_{Ai}, where i=1,...n.
Output: MGT, a generalization of PT[QI] with respect to k.

- Let the set of candidate nodes be C_i and the set of direct multi-attribute generalization relationships(edges) connecting these nodes by E_i. A modified breadth-first search is performed over the graph and we get a set of multi-attribute generalizations of size i.
- The set of candidate nodes of size (i+1) i.e. C_{i+1} and the edges E_{i+1} is constructed.
- Repeat the above steps until a node satisfying k-anonymity with respect to PT is found.
- Return the table with respect to the satisfying node as the MGT.

Birth Date	Sex	Zip Code	Disease
2/12/80	Male	110000	Fever
3/16/90	Female	110000	Ulcer
4/22/80	Male	110011	Cancer
5/23/80	Male	110011	Diabetes
6/1/90	Female	110012	Insomnia
10/2/80	Female	110012	Cough

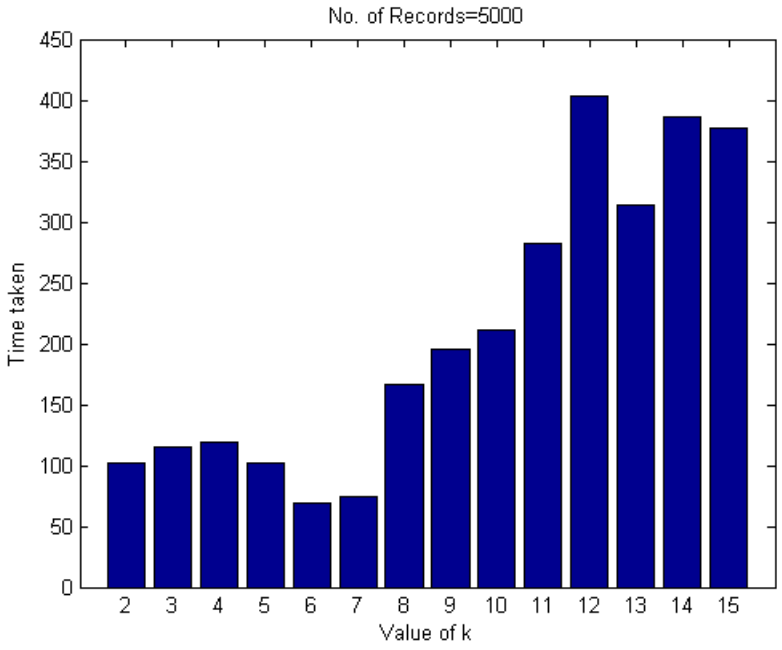
Hospital Patient Data

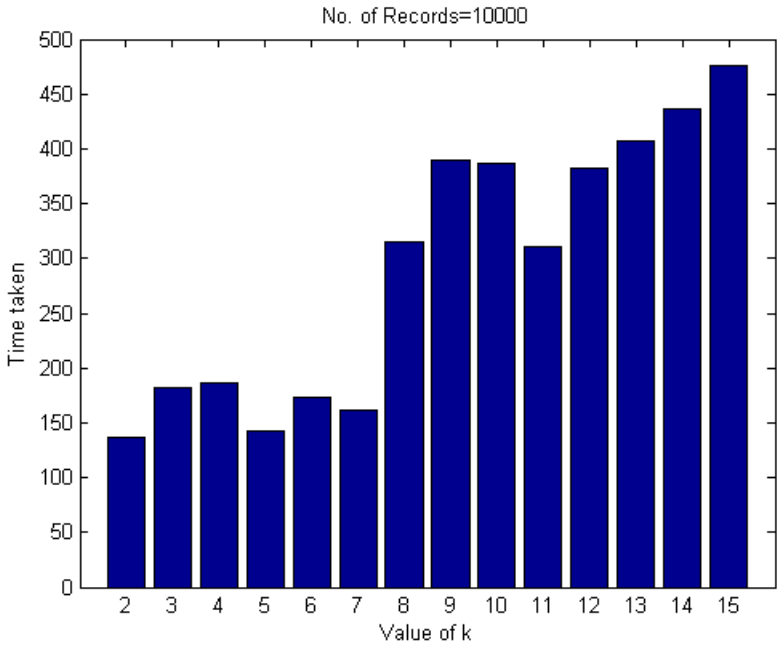


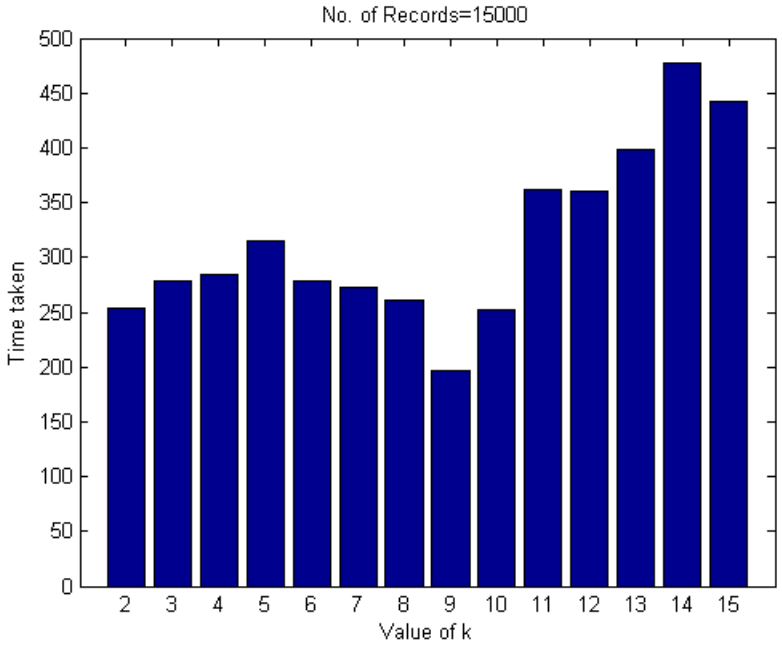
Domain Generalization Heirarchies (DGH) for different set of attributes

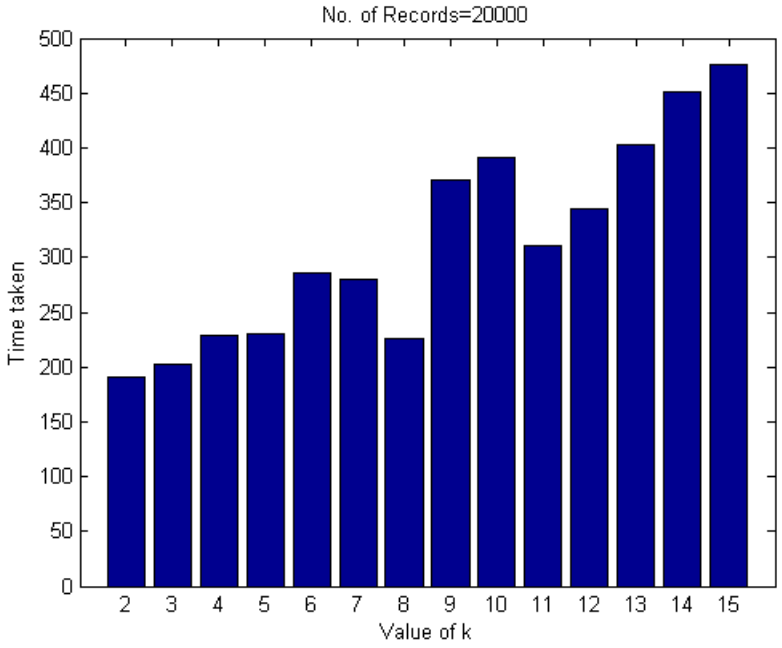
3.2.3 Results of Incognito Algorithm

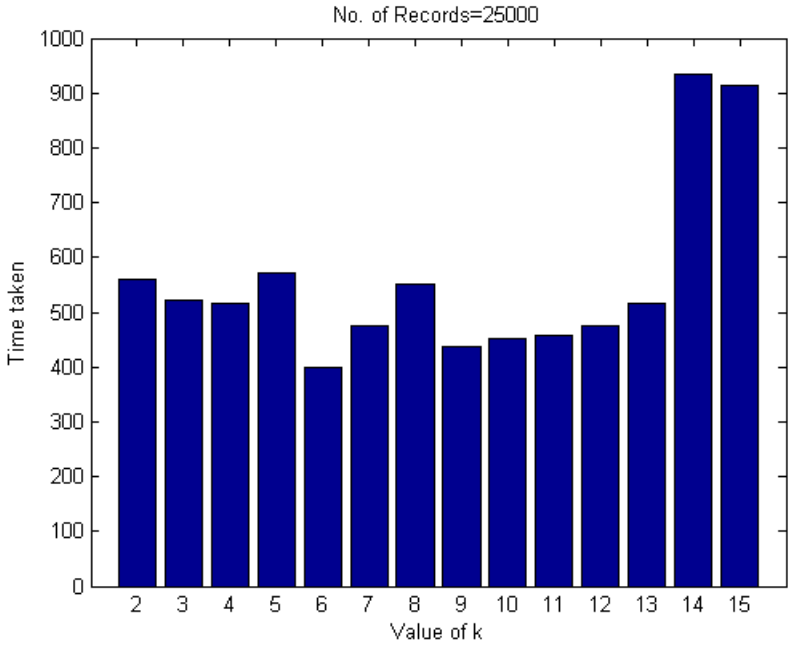
The basic incognito algorithm was applied to the adult dataset using java and oracle database. The java code was implemented in **eclipse juno** and for database **Oracle 10g** was used. The attributes **age,race,sex,education,marital-status** were taken as the quasi-identifier from the adult dataset.

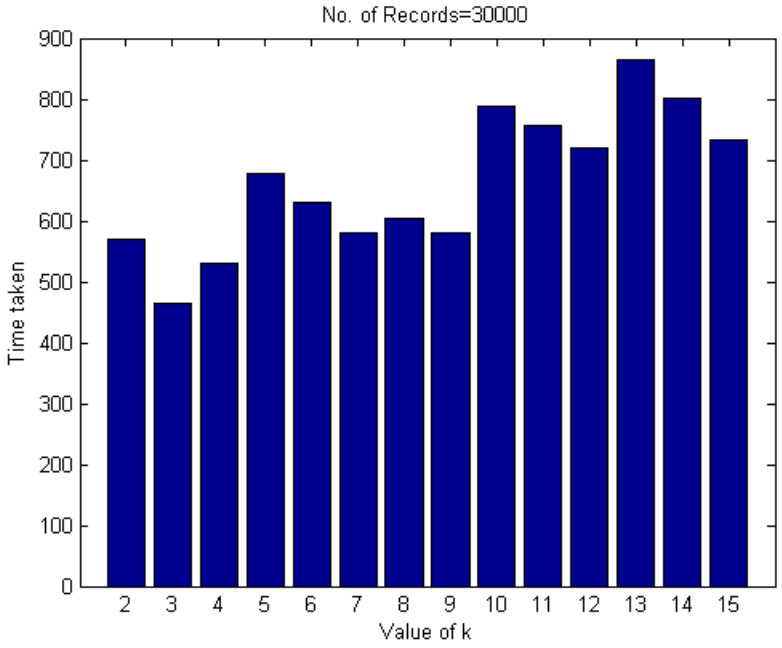





No. of Records=15000
Time taken
0
50
100
150
200
250
300
350
400
450
500
2 3 4 5 6 7 8 9 10 11 12 13 14 15
Value of k


No. of Records=20000
Time taken
0
50
100
150
200
250
300
350
400
450
500
2 3 4 5 6 7 8 9 10 11 12 13 14 15
Value of k


No. of Records=25000
Time taken
Value of k
0
100
200
300
400
500
600
700
800
900
1000
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=30000
Time taken
Value of k
0
100
200
300
400
500
600
700
800
900
2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.2.4 Advantages

- The algorithm finds all the k-anonymous full domain generalizations
- The optimal solution can be selected according to different criteria

3.2.5 Disadvantages

- The algorithm uses breadth first search method which takes a lot of time to traverse the solution space

3.3 Samarati's Algorithm

This algorithm searches for the possible k-anonymous solutions by jumping at different levels in Domain Generalization Hierarchy(DGH). It uses the binary search to obtain the solution in less time. This algorithm implements the AG_TS model. Therefore, suppression can be used to achieve k-anonymity. **MaxSup** is the maximum number of tuples that are allowed to be suppressed to achieve k-anonymity.

3.3.1 Generalized table - with suppression

Let T_i and T_j be two tables defined on the same set of attributes. Table T_j is said to be a generalization (with tuple suppression) of table T_i, denoted $T_i \leq T_j$, if:

- $| T_i | \leq | T_j$
- the domain dom(A, T_j) of each attribute A in T_j is equal to, or a generalization of, the domain dom(A, T_i) of attribute A in T_i
- it is possible to define an injective function associating each tuple t_j in T_j with a tuple t_i in T_i, such that the value of each attribute in t_j is equal to, or a generalization of, the value of the corresponding attribute in t_i

Race:R_0	ZIP:Z_0
asian	*94142*
asian	*94141*
asian	94139
asian	94139
asian	94139
black	*94138*
black	*94139*
white	*94139*
white	*94141*

(a)

Race:R_1	ZIP:Z_0
person	*94142*
person	94141
person	94139
person	94139
person	94139
person	*94138*
person	94139
person	94139
person	94141

(b)

Race:R_0	ZIP:Z_1
asian	9414*
asian	9414*
asian	9413*
asian	9413*
asian	9413*
black	9413*
black	9413*
white	*9413**
white	*9414**

(c)

Race:R_1	ZIP:Z_1
person	9414*
person	9414*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9414*

(d)

Race:R_0	ZIP:Z_2
asian	941**
asian	941**
asian	941**
asian	941**
asian	941**
black	941**
black	941**
white	941**
white	941**

(e)

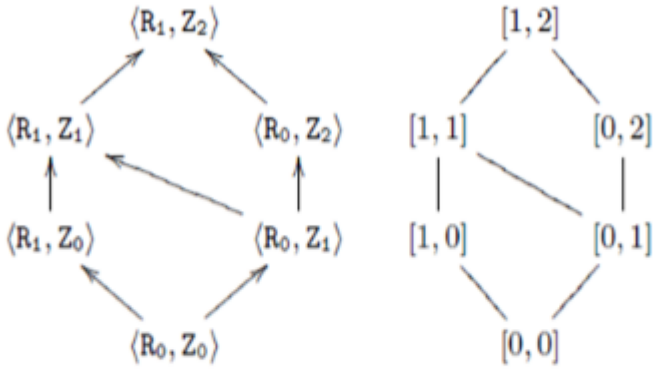
Race:R_1	ZIP:Z_2
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**

(f)

A private table PT (a) and its generalizations

3.3.2 Distance Vector

Let $T_i(A_1, ..., A_n)$ and $T_j(A_1, ...,A_n)$ be two tables such that $T_i \leq T$j . The distance vector of T_j from T_i is the vector $DV_{i,j} = [d_1, ..., d_n]$, where each d_z, z = 1, ...,n, is the length of the unique path between dom(A_z, T_i) and dom(A_z, T_j) in the domain generalization hierarchy DGH.



Domain Generalization Hierarchy DGH<R_0,Z_0> and corresponding hierarchy of distance vectors

Race:R_0	ZIP:Z_0
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Race:R_1	ZIP:Z_0
person	94141
person	94139
person	94139
person	94139
person	94139
person	94139
person	94141

$GT_{[1,0]}$

Race:R_0	ZIP:Z_1
asian	9414*
asian	9414*
asian	9413*
asian	9413*
asian	9413*
black	9413*
black	9413*

$GT_{[0,1]}$

A private table PT and its 2-minimal generalizations, assuming MaxSup=2

- The height of a distance vector DV in a distance vector lattice VL is denoted by height(DV, VL)
- if there is no solution that guarantees k-anonymity suppressing less than MaxSup tuples at height h, there cannot exist a solution, with height lower than h that guarantees it.
- This property is exploited by using a binary search approach on the lattice of distance vectors corresponding to the domain generalization hierarchy of the domains of the quasi-identier

3.3.3 k-minimal generalization - with suppression

Let T_i and T_j be two tables such that $T_i \leq T_j$, and let **MaxSup** be the specified threshold of acceptable suppression. T_j is said to be a k-minimal generalization of table T_i if the following conditions are satisfied:-

- T_j satisfies k-anonymity enforcing minimal required suppression, that is, T_j satisfies k-anonymity and $\forall T_z$: $T_i \leq T_z$;$DV_{i,z} = DV_{i,j}$; T_z satisfies k-anonymity $\Rightarrow | T_j | \geq | Tz |$

- $| T_i | - | T_j | \leq MaxSup$
- $\forall T_z$: $T_i \leq T_z$ and T_z satisfies conditions 1 and 2 $\Rightarrow \dashv (DV_{i,z}) < (DV_{i,j})$

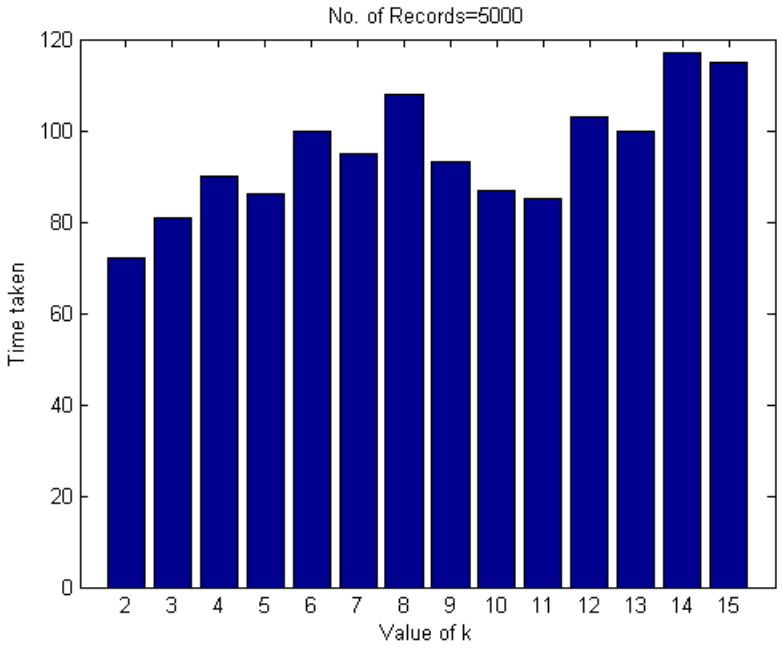
3.3.4 Algorithm

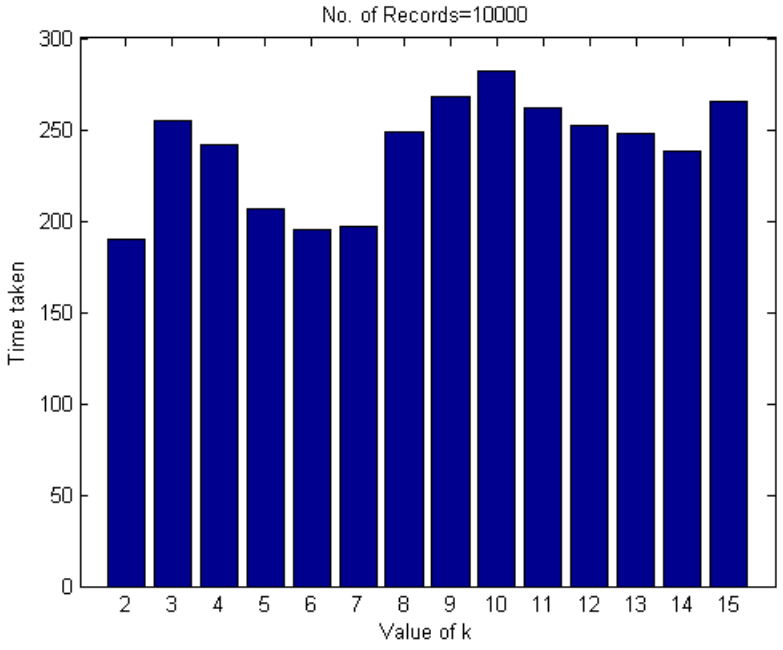
Input: Private table PT; quasi-identifier QI=(A_1,....,A_n), k constraint; lattice VL
Output: MGT, a generalization of PT[QI] with respect to k.

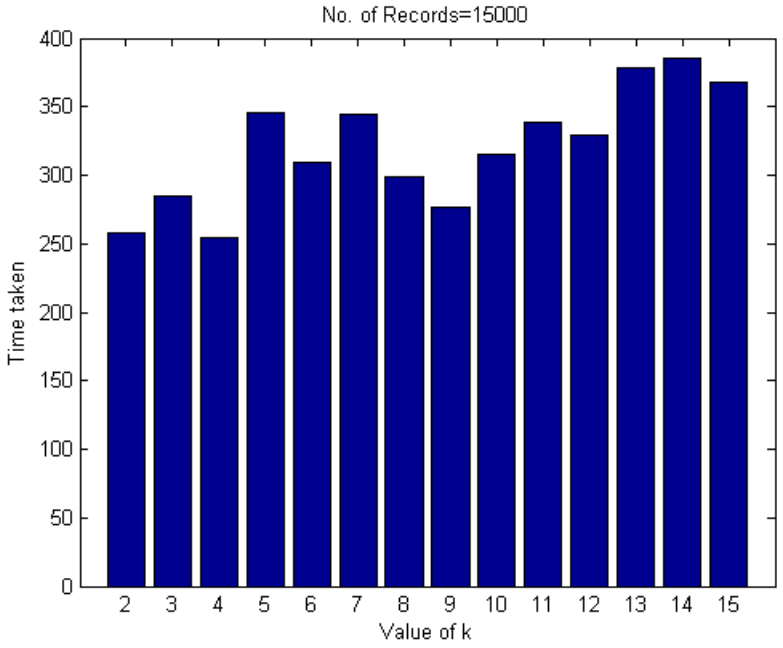
- Consider lattice VL of height h = height(T, VL), where T is the top element of the lattice.
- the vectors at height [h/2] are evaluated. If there is a vector that satisfies k-anonymity within the suppression threshold established at height [h/2], then:
  - the new area of search is the lower half i.e. from 0 to [h/4].
  - otherwise the new area of search is the upper half i.e. from [h/2] to [h].
- Repeat step 2 until the algorithm reaches the lowest height for which there is a distance vector that satisfies k-anonymity. Return the PT with respect to the distance vector as the MGT.

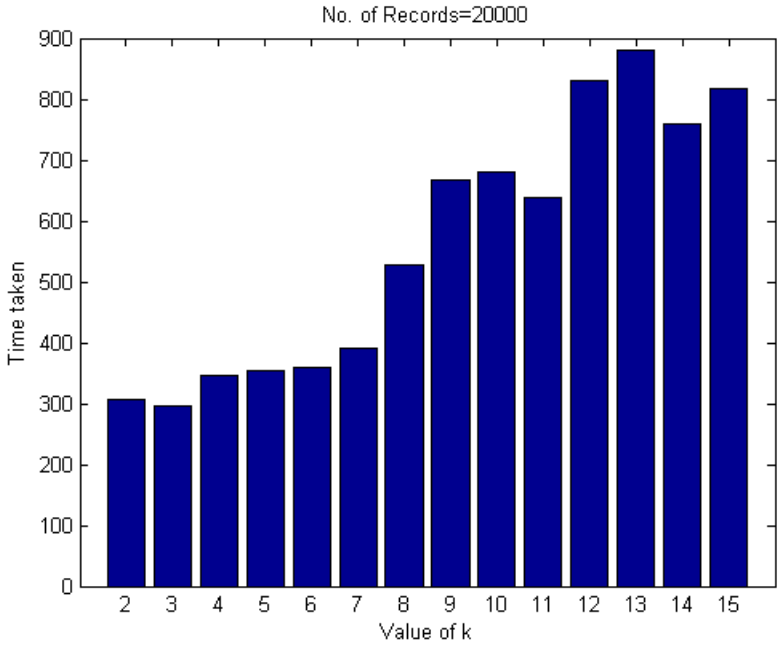
3.3.5 Results of Samarati's Algorithm

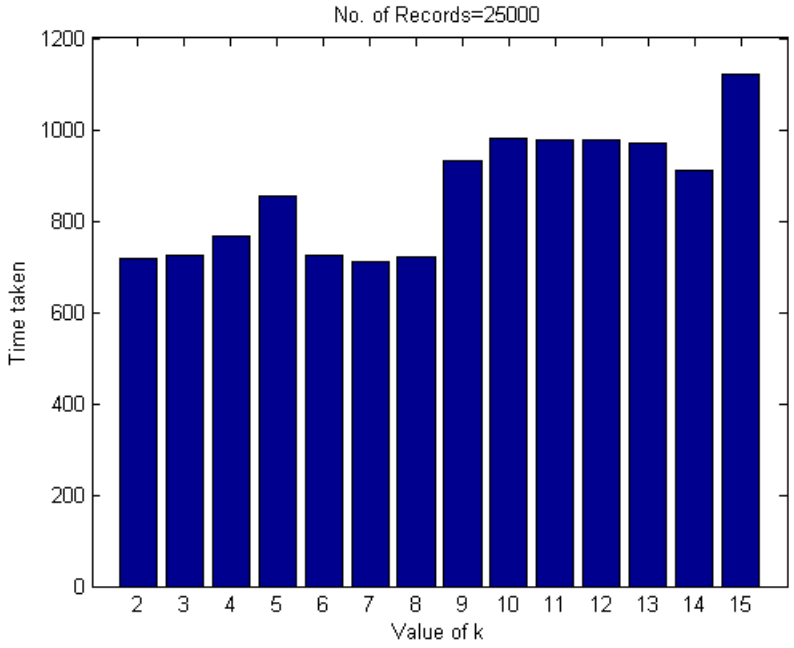
The Samarati's algorithm was applied to the adult dataset using java and oracle database. The java code was implemented in **eclipse juno** and for database **Oracle 10g** was used. The attributes **age,race,sex,education,marital-status** were taken as the quasi-identifier from the adult dataset.

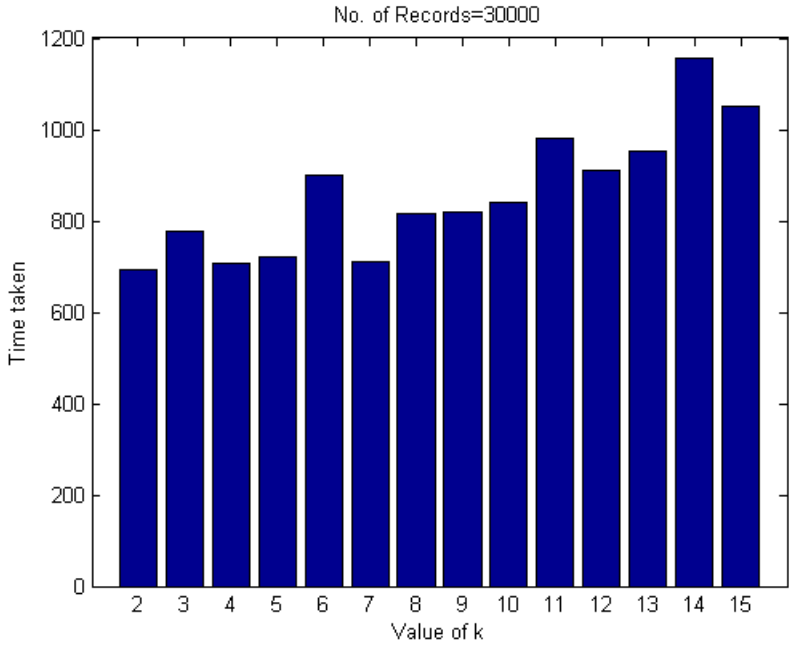





No. of Records=15000
Time taken
0
50
100
150
200
250
300
350
400
2 3 4 5 6 7 8 9 10 11 12 13 14 15
Value of k


No. of Records=20000
Time taken
0
100
200
300
400
500
600
700
800
900
2 3 4 5 6 7 8 9 10 11 12 13 14 15
Value of k


No. of Records=25000
Time taken
Value of k
0
200
400
600
800
1000
1200
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=30000
Time taken
Value of k
0
200
400
600
800
1000
1200
2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.4 Sweeney's Algorithm

According to Sweeney, the best solutions are attained after generalizing the variables with the unique values. The search space is the whole lattice. This approach only goes through a very small number of nodes in the lattice to find its solution. Thus, from a time perspective, this approach is very efficient.

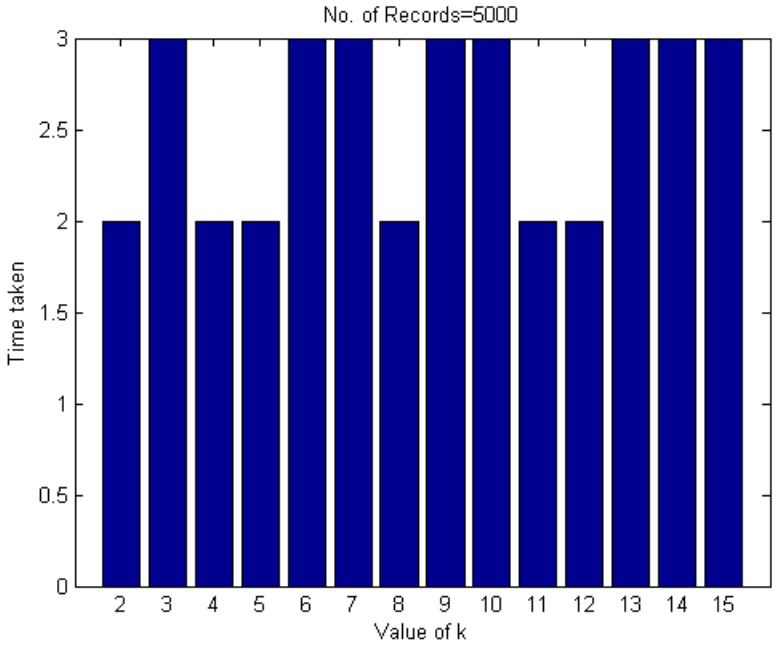
3.4.1 Algorithm

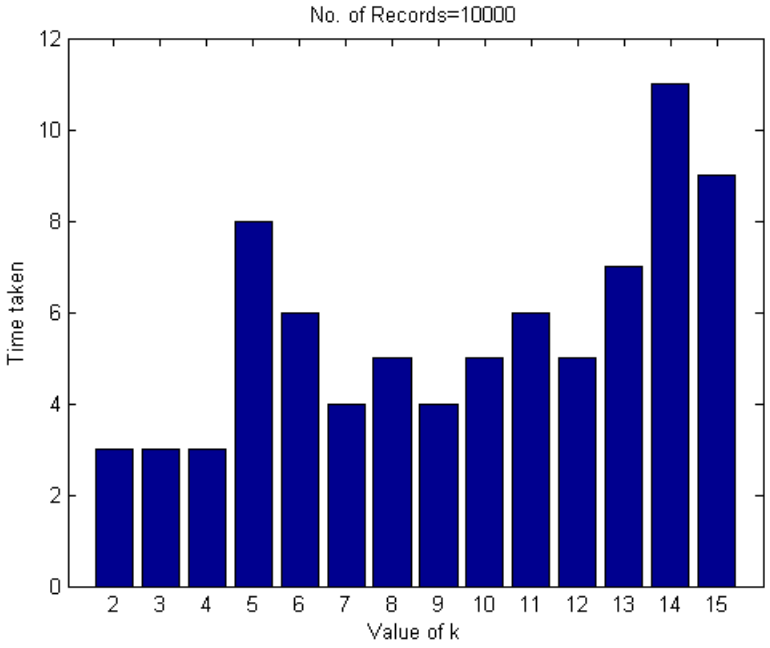
Input: Private table PT; quasi-identifier QI=(A_1,....,A_n), k constraint; hierarchies DGH_{Ai}, where i=1,...n.
Output: MGT, a generalization of PT[QI] with respect to k.

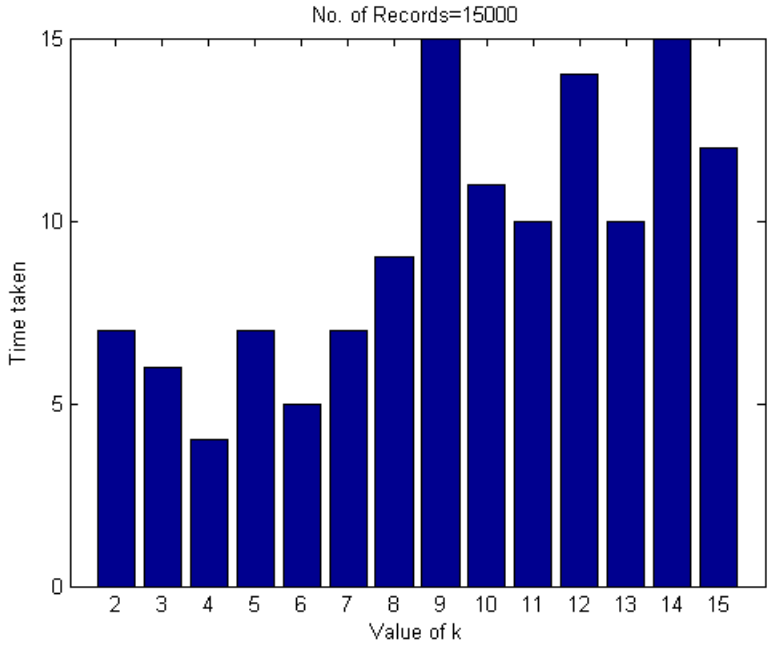
- Consider a table MGT = PT[QI]
- While k-anonymity is not achieved and the count of the remaining rows that do not comply to k-anonymity is more than k:
  - Get the number of distinct values of each attribute in MT
  - Generalize the attribute with the most distinct values
- Suppress the remaining rows and return MGT.

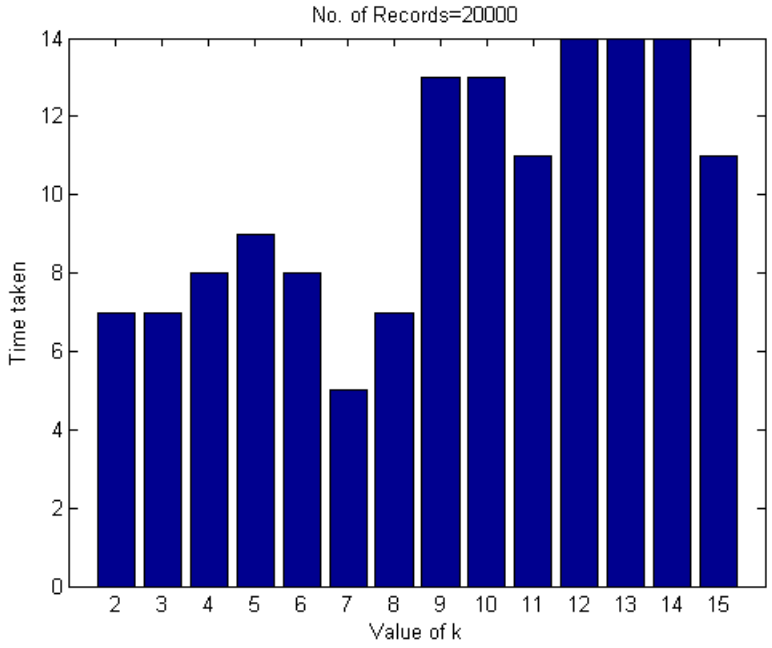
3.4.2 Results of Sweeney's Algorithm

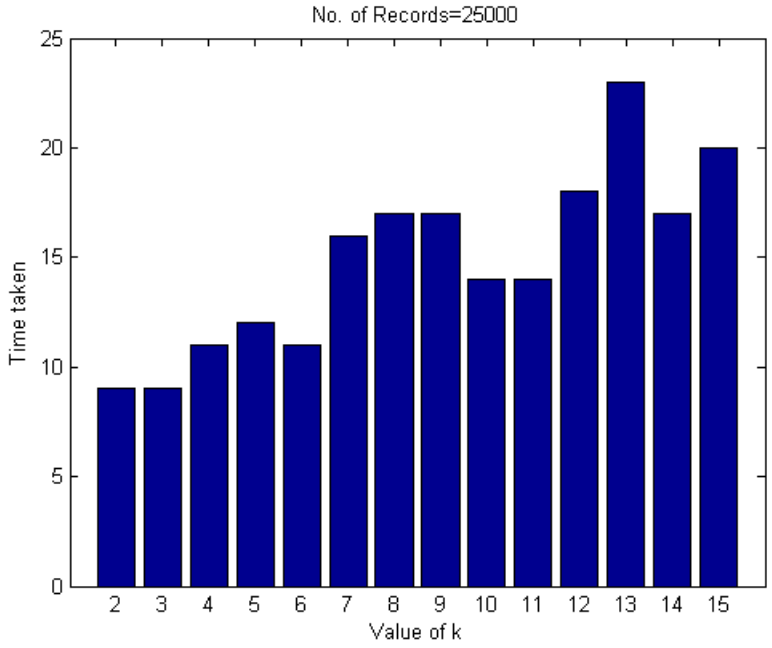
The Sweeney's algorithm was applied to the adult dataset using java and oracle database. The java code was implemented in **eclipse juno** and for database **Oracle 10g** was used. The attributes **age,race,sex,education,marital-status** were taken as the quasi-identifier from the adult dataset.

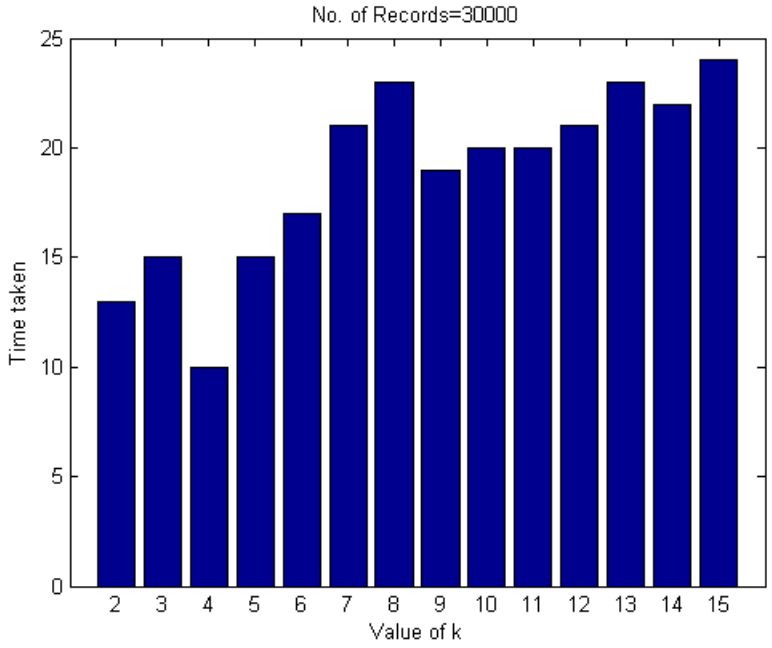

No. of Records=5000
Time taken
Value of k
0
0.5
1
1.5
2
2.5
3
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=10000
Time taken
Value of k
0
2
4
6
8
10
12
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=15000
Time taken
Value of k
0
5
10
15
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=20000
Time taken
Value of k
0
2
4
6
8
10
12
14
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=25000
Time taken
Value of k
0
5
10
15
20
25
2 3 4 5 6 7 8 9 10 11 12 13 14 15


No. of Records=30000
Time taken
Value of k
0
5
10
15
20
25
2 3 4 5 6 7 8 9 10 11 12 13 14 15

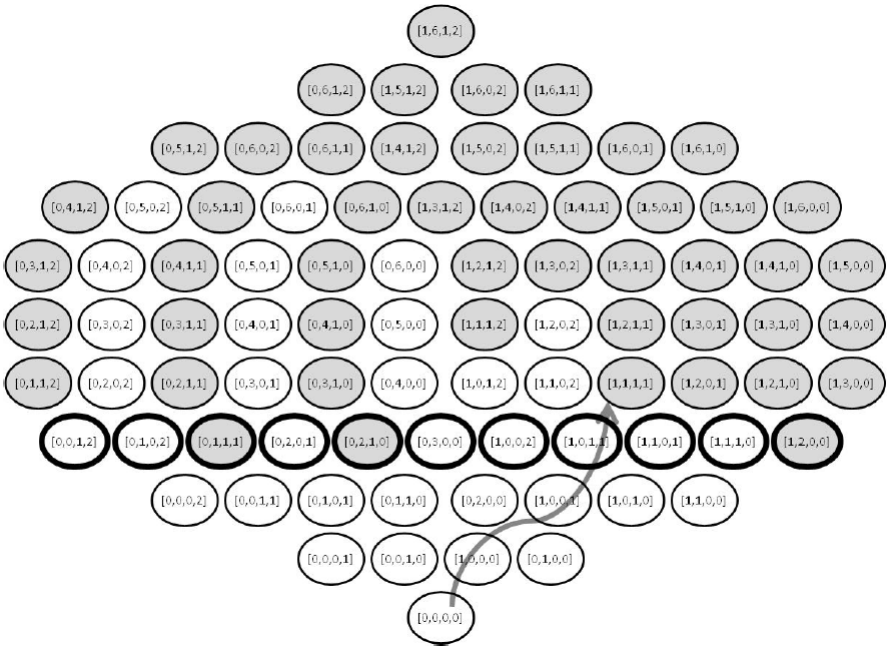
3.4.3 Advantage

- The algorithm checks very few nodes for k-anonymity due to which it is able to give results very fast.

3.4.4 Disadvantage

- The algorithm skips many nodes, therefore, the resulting data is very generalized and sometimes this released data may not be suitable for research purpose as it provides very little information.

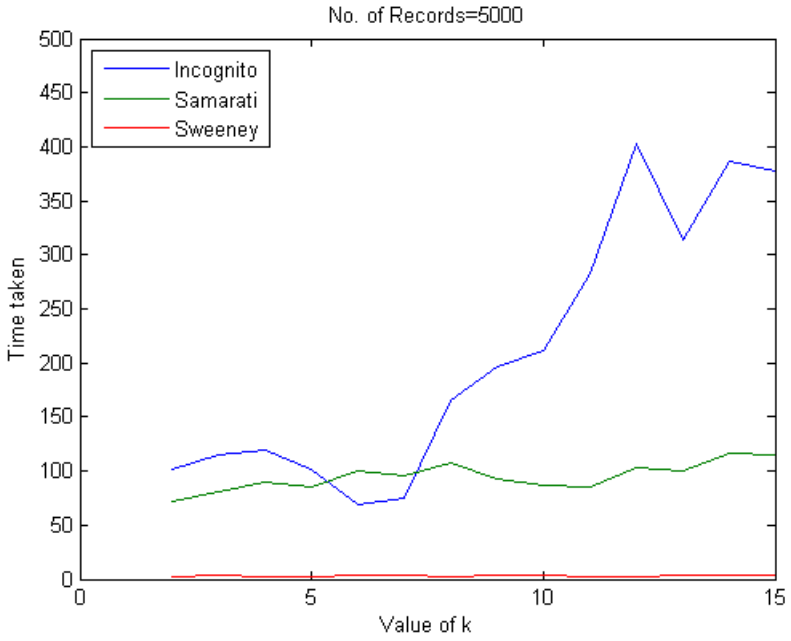
3.4.5 Comparison between Samarati's Algorithm and Datafly Algorithm

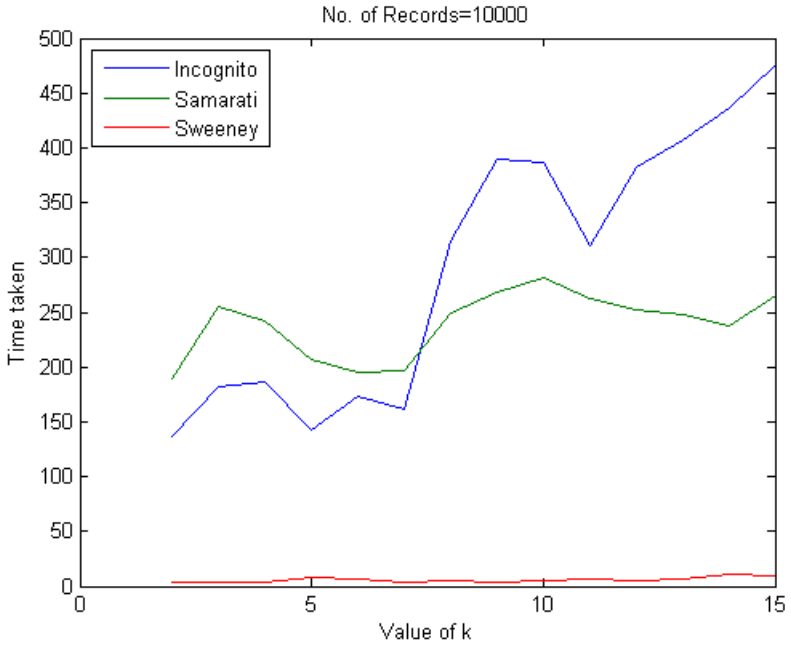


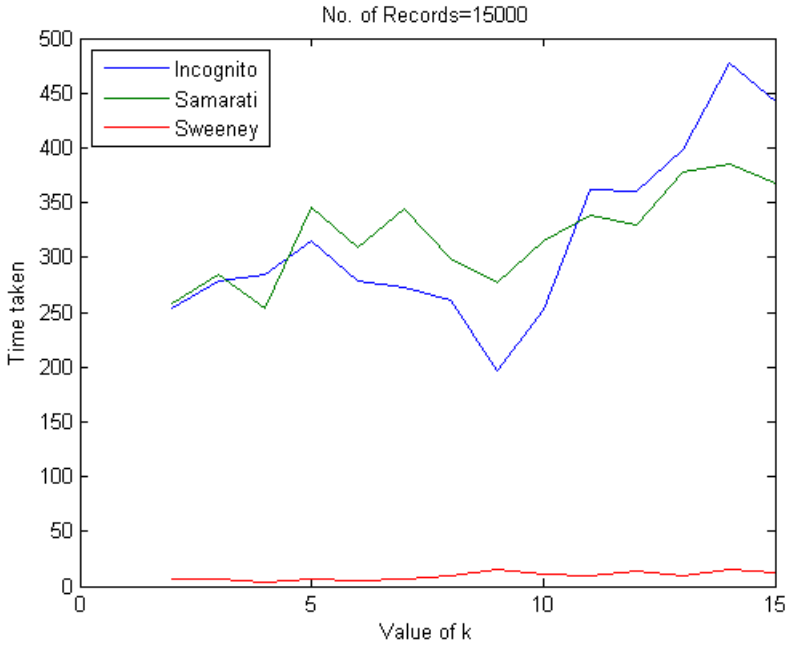
- The Samarati's algorithm evaluates all the nodes at a generalization level whereas Sweeney's algorithm skips a lot of nodes when moving between levels in search of a solution.
- The Samarati's algorithm provides a solution with minimal generalization and suppression which Sweeney's algorithm does not provide.
- Sweeney's algorithm is able to obtain a solution very fast as compared to Samarati's algorithm.

3.5 Results

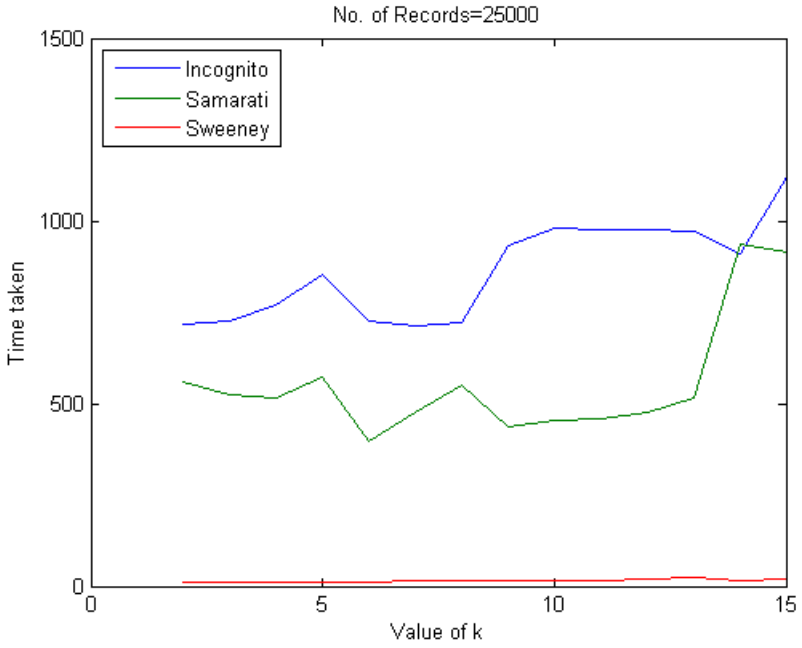
The result of all the algorithms combined together are shown below.We can clearly observe that as the value of k and number of records increase the time taken to achieve k-anonymity also increases.

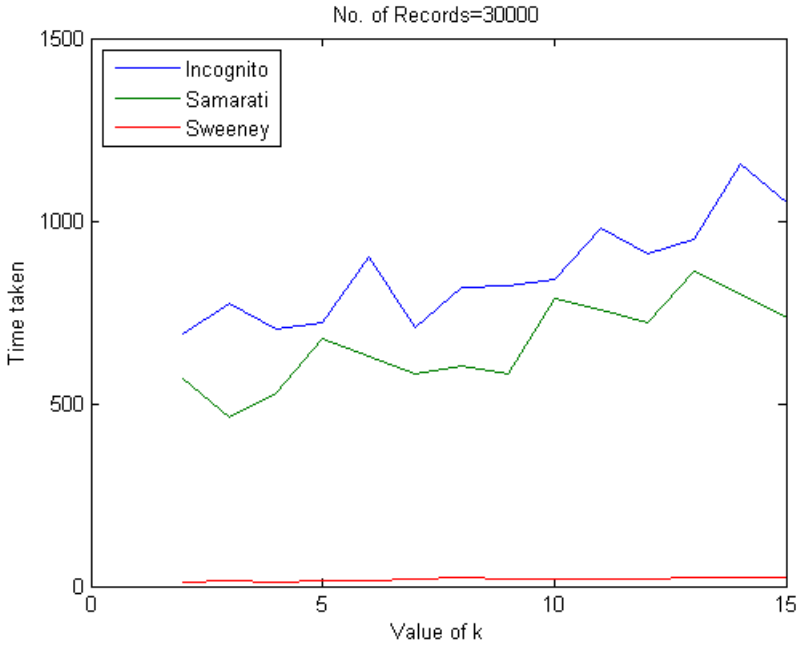





No. of Records=15000
Incognito
Samarati
Sweeney
Time taken
Value of k
0
50
100
150
200
250
300
350
400
450
500
0
5
10
15


No. of Records=20000
Incognito
Samarati
Sweeney
Time taken
Value of k
0
100
200
300
400
500
600
700
800
900
1000
0
5
10
15


No. of Records=25000
Incognito
Samarati
Sweeney
Time taken
Value of k
0
500
1000
1500
0
5
10
15


No. of Records=30000
Incognito
Samarati
Sweeney
Time taken
Value of k
0
500
1000
1500
0
5
10
15

Chapter 4

Conclusion and Future Work

In this thesis we described how microdata released by various organisations for research or business purpose can compromise the security and privacy of an individual. In order to guarantee the anonymity of individuals and protect the released microdata from any attacks we discussed the work that has been done in order to protect the released microdata by the means of k-anonymity. Three basic algorithms for k-anonymity, Incognito algorithm, Samarati's algorithm and Sweeney's algorithm, were studied upon where each of the algorithms had certain advantages and disadvantages. These algorithms were based on the AG_ model and AG_TS model of k-anonymity. We measured the effectiveness of these algorithms by plotting the graph between value of k and time taken to achieve k-anonymity with respect to the number of records. These algorithms have been really helpful in reducing the number of attacks on the microdata and securing sensitive information of the individuals.

4.0.1 Future Work

The algorithms discussed in this thesis can be further improved by reducing the size of the solution space and applying improved searching algorithms. An effective algorithm can be applied by combining the Samarati's algorithm and Sweeney's algorithm. The Sweeney's algorithm can be used to reduce the solution space by achieving the lower limits and upper limits effectively for the Samarati's algorithm and then Samarati's algorithm can be applied to obtain the efficient solution for k-anonymity. This technique can greatly reduce the time taken to achieve k-anonymity and also produce an efficient solution

Bibliography

1. **Latanya Sweeney**. *Achieving k-anonymity Privacy Protection using Generalization and Suppression.* International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5):571-588, 2002.

2. **Romeo Issa.** *Satisfying k-anonymity:New Algorithm and Empirical Evaluation*, Master of Computer Science, Carleton University. M.Sc Thesis, January 2009.

3. **V. Ciriani, S. De Capitani di Vimercati, S. Foresti and P. Samarati**, *k-Anonymity*, Università degli Studi di Milano, 26013 Crema, Italia.

4. **P. Samarati and L. Sweeney**. *Protecting Privacy when Disclosing Information: k-anonymity and its enforcements through generalization and suppression.* Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998

5. **Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan**. *Incognito:Efficient Full-Domain K-anonymity*, University of Wisconsin, Madison

6. **Dr Kerina Jones**. *Personal identity protection solutions in the presence of low copy number fields*, HIRU, Swansea University. EUCONET Record Linkage Workshop, 15^{th} to 17^{th} June 2011.

7. **Pierangela Samarati.** *k-anonymity.* Dipartmento di Tecnologie dell'Informazione Universit´a degli Studi di Milano, FOSAD 2008.